

# Autonomous Apple Fruitlet Sizing and Growth Rate Tracking using Computer Vision

Harry Freeman <sup>1\*</sup>, Mohamad Qadri <sup>1</sup>, Abhisesh Silwal <sup>1</sup>, Paul O'Connor <sup>2</sup>, Zachary Rubinstein <sup>1</sup>, Daniel Cooley <sup>2</sup>, and George Kantor <sup>1</sup>

**Abstract**—Measuring growth rates of apple fruitlets is important because it allows apple growers to determine when to apply chemical thinners to their crops to optimize yield. The current practice of obtaining growth rates involves using calipers to record sizes of fruitlets across multiple days. Due to the number of fruitlets needed to be sized, this method is laborious, time-consuming, and prone to human error. In this paper, we present a computer vision approach to measure the sizes and growth rates of apple fruitlets. With images collected by a hand-held stereo camera, our system detects, segments, and fits ellipses to fruitlets to measure their diameters. To measure growth rates, we utilize an Attentional Graph Neural Network to associate fruitlets across different days. We provide quantitative results on data collected in an apple orchard, and demonstrate that our system is able to predict abscise rates within 3% of the current method with a 7 times improvement in speed, while requiring significantly less manual effort. Moreover, we provide results on images captured by a robotic system in the field, and discuss the next steps to make the process fully autonomous.

**Index Terms**—Agricultural Automation, Computer Vision for Automation, Robotics in Agriculture and Forestry, Deep Learning in Robotics and Automation

## I. INTRODUCTION

Recent advancements in computer vision have allowed farmers to deploy autonomous plant monitoring solutions to more efficiently inspect vast quantities of crops. Computer vision systems provide fast and reliable information for downstream tasks such as harvesting [1], [2], phenotyping [3], [4], and yield prediction [5]–[7], ultimately allowing farmers to make real-time crop management decisions. Common computer vision applications in agriculture include mapping [8], [9], counting [10]–[12], modelling [13], [14], and disease classification [15].

We focus on the target application of measuring growth rates of apple fruitlets. This is important because it enables farmers to better control their annual yield. It is standard practice to thin apple trees to prevent them from developing a pattern of alternative year bearing in order to produce a more consistent yearly harvest. To predict the effect of thinning application applied on trees, the Fruitlet Growth Model developed by Greene *et al.* [16] is used to determine how often farmers need to spray. The model takes into account the growth rates of a subset of fruitlets over multiple days. The fruitlets with growth rates greater than 50% of the fastest growing

fruits are predicted to persist the thinning. Farmers calculate the percentage expected to abscise and use this information to determine when to apply thinning application. Therefore, increasing the sampling size and producing more consistent measurements will lead to more accurate and reliable yield predictions.

The sizing method most commonly used in practice involves identifying each individual fruitlet, using a digital caliper to hand-measure sizes, and manually entering the data into a spreadsheet so that growth rates can be tracked. We will refer to this sizing process as the caliper method. The number of fruitlets typically sized using the caliper method lies in the range of high hundreds to low thousands. Sizing is performed twice on each fruitlet per thinning application: once three to four days after application, and again seven to eight days after application. Taking this many hand-measurements is not only labor-intensive, but highly subject to human error. It is inefficient and time-consuming for a human to record caliper readings for hundreds to thousands of fruitlets across multiple days, making some farmers hesitant to adopt the approach. Manually associating fruitlets across different days is also very challenging; fruitlets are likely to have moved or fallen off, resulting in them being mis-identified which negatively affects growth estimates. Moreover, using calipers creates variability when measuring asymmetrically shaped fruit, leading to inaccurate measurements which become more pronounced as different workers are employed to collect data. As a result, there is a need to automate this process to make sizing faster, more repeatable, and more accurate.

There has been significant work dedicated towards sizing fruit in agriculture. In the work of [6], calibration spheres are placed on trees and used as reference scales to estimate the sizes of segmented apples. Similarly, Wang *et al.* [17] are able to size fruits in the field with a smart phone by placing a reference circle of known size behind the fruit. While these methods only require simple segmentation and sizing algorithms, they do not extend well to fruitlets as it is impractical to place reference objects behind every fruit in such occluded environments.

Approaches have been developed to size fruits in 3D. Reconstruction-based methods are used by [18], [19] where 3D models are created from multiple sensor measurements. However, these methods are computationally expensive and do not perform well with occlusions where reconstructions are often incomplete. To address these issues, automated shape completion methods have been implemented by [20], [21] which fit superellipsoids to accumulated point clouds. These

<sup>1</sup>Carnegie Mellon University Robotics Institute, PA, USA {hfreeman, mqadri, asilwal, zbr, gkantor}@andrew.cmu.edu

<sup>2</sup>University of Massachusetts Amherst Stockbridge School of Agriculture, MA, USA {proconnor, dcooley}@umass.edu

methods either rely on successive frame alignment algorithms, such as Iterative Closest Point, which fail in agricultural environments due to the dynamic structure of plants, or use expensive raycasting operations which are slow when performed at finer resolutions. Non-reconstruction based 3D sizing is performed by [22] where the major-axis of an apple is fit to 3D points collected from a single camera image and time-of-flight sensor. However, the performance is poor, achieving an accuracy of 69.1%. The work of [23] also sizes apples from single images by fitting spheres in 3D. This would not adapt well to apple fruitlets due their small size and asymmetrical shape.

As an alternative to 3D sizing, 2D photogrammetric methods have been adopted in agriculture. These methods either directly estimate the widths of fruits [22] or fit ellipses to measure the heights and widths [24]–[26]. This is advantageous as sizes are able to be extracted fast from 2D images without the need to aggregate information from multiple views. However, the presented approaches use simple detection and segmentation modules, relying on color information and non-deep features. While these methods work well in their respective domains, they would fail when trying to detect and segment fruitlets. This is because the proximity of fruitlets are much closer together and their colors blend in with the surrounding leaves.

There has been limited work on cross-day fruit association, which is required to measure growth rates. While there has been effort dedicated towards both fruit tracking [9], [27] and identification from different camera views [28], these methods either rely on images taken from consecutive frames or maintaining 3D knowledge about the current scene. Hondo *et al.* [29] develop a deep learning approach to size apples and track their growth over time. However, images are captured from a fixed camera pose, and fruit identification is performed by comparing center coordinates of segmented apples. This is infeasible for fruitlets due to their small size, dynamic nature, close proximity, and the number of clusters needed to be sized.

In this paper, we present a computer vision-based system for sizing and measuring the growth rates of apple fruitlets using single stereo image pairs generated from a hand-held camera. To automate the measurement of growth rates, we develop a novel Graph Neural Network (GNN) [30] approach for cross-day fruitlet association, which, to the best of our knowledge, is the first of its kind used in agriculture. We demonstrate that the results produced by our method are comparable to those produced by the caliper method, with the ability to reduce the labor-intensive effort and significantly improve speed. The specific contributions of this paper are:

- i A computer vision-based systems approach to detect, segment, and size apple fruitlets
- ii An Attentional Graph Neural Network approach for cross-day fruitlet association
- iii Experiments and results on data collected by a hand-held camera in an apple orchard
- iv Quantitative evaluation on data collected by a robotic system in an apple orchard

## II. METHOD

### A. Tagging Methodology

The Fruitlet Growth Model requires growers to determine the number of trees and the number of clusters per tree they want to size. A cluster is a group of typically two to six fruitlets that grow in near-proximity to one another. The diameters of each fruitlet in every selected cluster are measured two to three times per thinning application.

We hang AprilTags [31] next to each cluster to identify the selected fruitlets. AprilTags were selected because they allow for fast cluster identification in computer vision systems. They are easy to detect and segment, and can be used as additional semantic information for fruitlet association as demonstrated in Section II-C2. Each fruitlet in the cluster is assigned a unique id for identification and tracking its size (Fig. 1).



Fig. 1. An example of a tagged fruitlet cluster. An AprilTag is hung next to the cluster, and each fruitlet receives a unique id which is written on the back for identification.

### B. Sizing

Images are captured using a hand-held flash stereo camera [32] as seen in Fig. 2. Stereo camera systems are commonly used in agriculture to extract 3D information as a result of their reliability and low cost. They resolve finer details compared to systems that use LiDAR [33]–[35], and flash-based system are more resilient to varying illumination conditions where RGB-D sensors inconsistently perform [22], [24].

To facilitate data collection, we designed a custom setup that consists of a stereo camera connected via USB to a laptop to save the captured images. A phone is mounted on the back and connected via USB-C to the same laptop to allow the user to visualize and assess the quality of the images in real-time.

Our apple fruitlet sizing pipeline is based on that presented by Qadri [36]. Fruitlets are detected, segmented, and sized by fitting an ellipse. The diameter of each fruitlet is calculated using the minor axis of the fit ellipse and the extracted disparity values. An overview of our system is presented in Fig. 3.

The main difference in our approach is we replace the MADNet [37] network used by [36] with RAFT-Stereo [38]. RAFT-Stereo is a state-of-the-art deep learning-based stereo matching network that outperforms traditional disparity generation methods, such as SGBM [39], and does not require fine-tuning. This is advantageous as the network does not need to

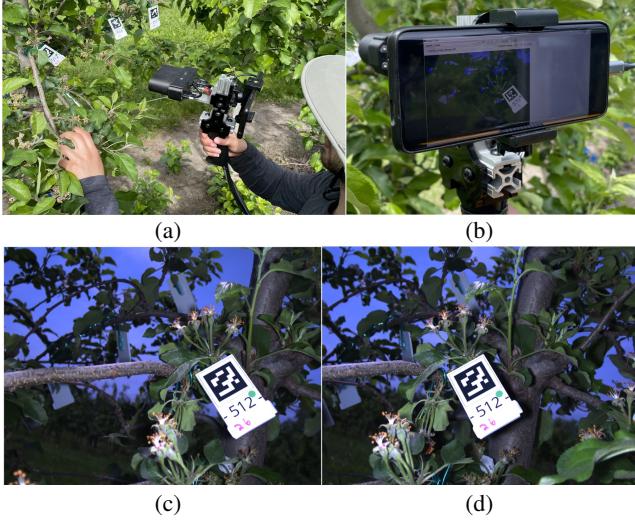
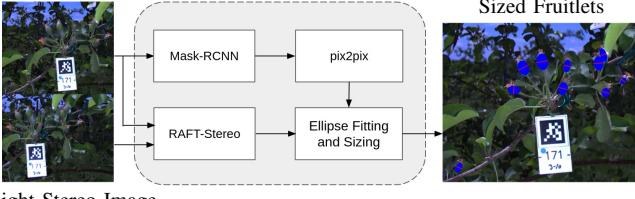


Fig. 2. (a) Hand-held flash stereo camera. (b) Phone is mounted and connected via USB-C to display images to the user in real-time. Captured left (c) and right (d) stereo images are shown.

be retrained between datasets or when switching out the stereo camera being used. We also replace the Faster-RCNN [40] network with Mask-RCNN [41] to detect and predict bounding boxes around all fruitlets in the image. This is because Mask-RCNN can be simultaneously used for tag segmentation, as discussed in Section II-C2, without requiring an additional network pass. Only the bounding box classification and regression heads are used for fruitlet detection, and the mask segmentation head is ignored. We use a customized detectron2 [42] Mask-RCNN implementation.

Left Stereo Image



Right Stereo Image

Fig. 3. Fruitlet sizing pipeline. Fruitlets are detected using Mask-RCNN bounding box classification and regressions heads and segmented using pix2pix. Ellipses are fit to the segmented fruitlets and sized using disparity values extracted by RAFT-Stereo.

To segment, we train a pix2pix [43] Conditional Generative Adversarial Network (CGAN). Each individual bounding box is cropped and passed to the pix2pix generator, which outputs a single segmentation mask for the fruitlet (Fig. 4).

The reason the proposed detection and segmentation networks are used instead of a single instance segmentation Mask-RCNN network is because of the challenges of obtaining labelled ground-truth data, which is a common issue in agriculture [44]. Hand-segmenting fruitlets for the required number of images to train is a time-consuming task, and is difficult to outsource as a result of the required domain-specific knowledge. It is more data-efficient to train a network to semantically label individual fruitlets [36]. A CGAN architecture was chosen as the segmentation network because CGANs have demonstrated previous success in operating in

agricultural environments [36], [45], [46]. As well, pix2pix performed better than Mask-RCNN for fruitlet segmentation when trained on low amounts of labelled data.

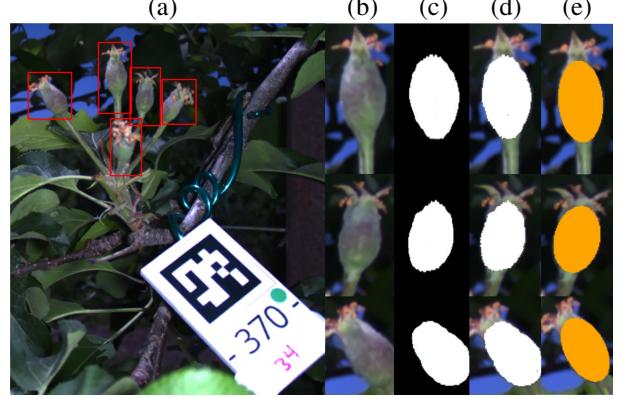


Fig. 4. Segmentation of apple fruitlets. Detected fruitlets (a) are cropped (b) and passed to the pix2pix generator which outputs a segmentation mask (c) and (d) to be used for ellipse fitting (e).

An ellipse is fit to the segmented image following the process demonstrated in Fig. 5. First, a binary threshold is applied to the pix2pix output. Next, the contour surrounding the segmented points is extracted. Lastly, an ellipse is fit using the OpenCV [47] fitEllipse function, which uses a constrained least squares formulation. The result is each ellipses' canonical parameters, including the length of the minor axis  $ma$ . The size is calculated as

$$\text{size} = \frac{ma \times b}{d} \quad (1)$$

where  $b$  is the baseline of the stereo camera and  $d$  is the max disparity value found in a square region around the center of the segmented fruitlet. The derivation of Equation 1 can be found in [36].

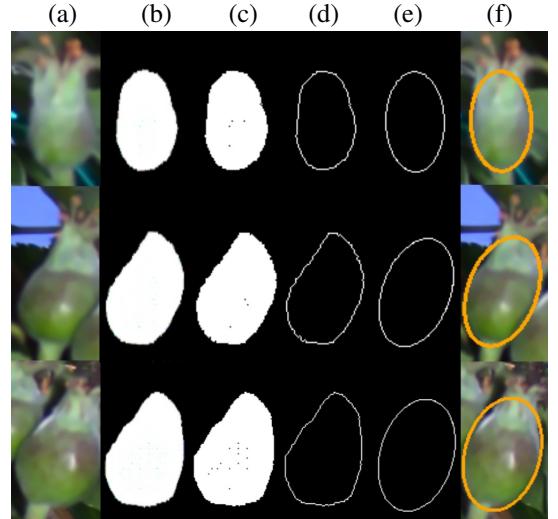


Fig. 5. Fruitlet ellipse fitting. The pix2pix output (b) is thresholded (c) and a contour is fit around the segmented image (d). An ellipse is fit using the OpenCV fitEllipse function to produce (e) and (f).

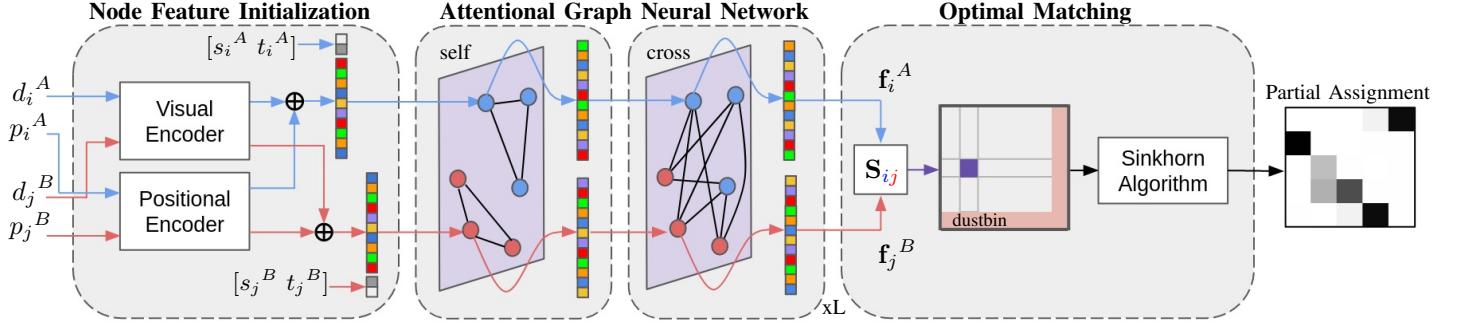


Fig. 6. Cross-day fruitlet association network architecture. Local features are mapped to deep vectors using visual and positional descriptor encoders, and the result is concatenated with node classification scores and tag information to build the initial node feature vectors. The feature vectors are updated through a series of  $L$  alternating self and cross attention layers, and the result is passed through an optimal matching layer to find the optimal partial assignment.

### C. Cross-Day Fruitlet Association

While the system presented in Section II-B helps automate the fruitlet sizing process by removing the need for hand-caliper measurements, very little to no research has been done to compare sizes across different days. As a result, human-effort is still required to determine which fruitlets in one image match to fruitlets in the other so that growth rates can be properly calculated. This task is time-consuming and requires great attention to detail. If we wish to move towards full automation of the fruitlet sizing process, alternative solutions must be found.

It is challenging to accurately associate fruitlets in images taken on different days. This is because fruitlets may have moved, fallen off, or changed appearance, and the images are unlikely to have been taken from the same camera pose. To address these issues, we introduce a GNN-based system for cross-day fruitlet association. Given two images of the same cluster taken on different days, the task is to identify the fruitlets belonging to the cluster in one image with those in the other. We use a GNN over traditional Convolutional Neural Network (CNN) based architectures because CNNs assume the input to be a grid-like structure where convolutions are performed on neighbouring pixels. This assumption does not accurately represent fruitlet association as there is no guarantee of spatial locality between fruitlets, especially across images. GNNs have shown impressive results when operating on unstructured data, where node feature embeddings are propagated along edges in a process known as feature aggregation. They have successfully been applied to multi-object tracking tasks [48], [49] and feature matching from different camera poses [50], [51]. There have been several GNN architectures introduced [52]–[54] each with their own unique feature aggregation methods designed for their desired tasks. Particularly, Graph Attention Networks (GATs) [55] have gained a lot of traction as they are more generalizable and are able to employ attention mechanisms in order to assign importance to neighbouring nodes without requiring knowledge about the graph structure upfront.

To perform cluster association, we utilize an Attentional Graph Neural Network, a type of GAT introduced by Sarlin *et al.* [50]. The main difference in our problem formulation and the one in [50] is that instead of matching keypoints

consisting of single pixels, we are matching fruitlet detections which span multiple pixels. The novelty in our approach lies in i) our local feature extractor (Section II-C3) which extracts visual and positional descriptors from fruitlet and tag detections; ii) our node feature vector initialization (Section II-C4), which encodes the local features using CNNs and directly injects classification score and tag information onto the resulting feature vector; and iii) our mechanism to apply loss to exclusively clustered fruitlets (Sections II-C2 and II-C8) due to the difficulties in labelling ground-truth data. The cross-day fruitlet association network can be seen in Fig. 6.

1) *Formulation:* Consider images  $A$  and  $B$  taken on different days with detected fruitlets  $\mathbf{F}^A$  and  $\mathbf{F}^B$  belonging to the same cluster  $C$ .  $\mathbf{F}^A$  and  $\mathbf{F}^B$  have  $M$  and  $N$  fruitlets respectively, index by  $\mathcal{A} := \{1, \dots, M\}$  and  $\mathcal{B} := \{1, \dots, N\}$ . The objective is to match fruitlets in  $\mathbf{F}^A$  with fruitlets in  $\mathbf{F}^B$ . Similar to [50], each fruitlet must adhere to a set of constraints: i) must have at most a single correspondance in the other image; ii) may be unmatched as a result of fruitlets falling off, occlusions, missed detections, or incorrectly detected fruitlets.

2) *Detection, Segmentation, and Feature Map Extraction:* An image  $I \in \{A, B\}$  of target cluster  $C$  is passed through a Mask-RCNN network to detect fruitlet bounding boxes  $\mathbf{B}^I$  and detect and segment tags  $\mathbf{T}^I$ . As in Section II-B, only the bounding box classification and regression heads are used to detect fruitlets, whereas the mask segmentation head is also used for tag segmentation. An additional class output is added to the network to identify fruitlets that belong to the imaged cluster. In other words, the network identifies the clustered fruitlets  $\mathbf{F}^I$  by determining which  $b_i \in \mathbf{B}^I$  belong to  $C$ . This is necessary for three reasons: first, the Fruitlet Growth Model requires only the fruitlets belonging to the tagged cluster to be measured; second, our recorded caliper data to evaluate against only contains measured growth rates for clustered fruitlets; and third, for network training it is infeasible to accurately label fruitlet matches for every fruitlet in hundreds of images, and is it much more labour and time-efficient to focus on only clustered fruitlets. An example of the Mask-RCNN and AprilTag identification output can be seen in Fig. 7.

Detected fruitlets are segmented using pix2pix, and tag  $\tau \in \mathbf{T}^I$  associated with  $C$  is identified by AprilTag id. The resulting feature maps from the ResNet-101 [56] Feature

Pyramid Network (FPN) [57] are extracted to be used for local feature extraction.

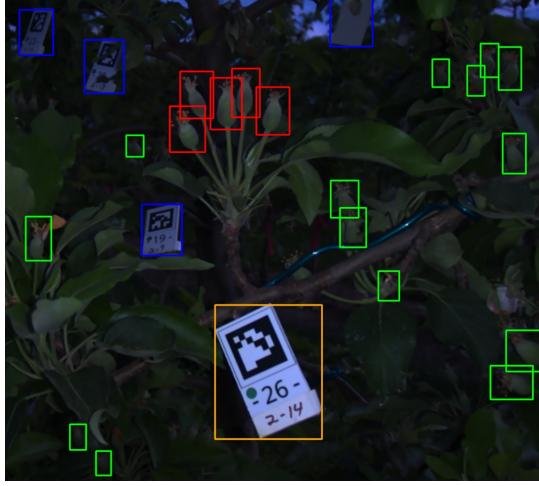


Fig. 7. Example output from Mask-RCNN network and AprilTag detector. Fruitlets are classified as cluster (red) or non-cluster (green). The cluster tag (orange) is identified by AprilTag id.

3) *Local Feature Extraction:* Contrary to [50] whose local features are a combination of visual descriptors and keypoint positions, we embed additional disparity and segmentation information onto the keypoints. As a result, our local features consist of both visual and positional descriptors. We denote  $\mathbf{A}^I = \{\mathbf{B}^I \cup \tau\}$  as the association set. For an  $a_i \in \mathbf{A}^I$  with visual descriptor  $\mathbf{d}_i$  and positional descriptor  $\mathbf{p}_i$ , the local feature is the combination  $(\mathbf{d}_i, \mathbf{p}_i)$ .

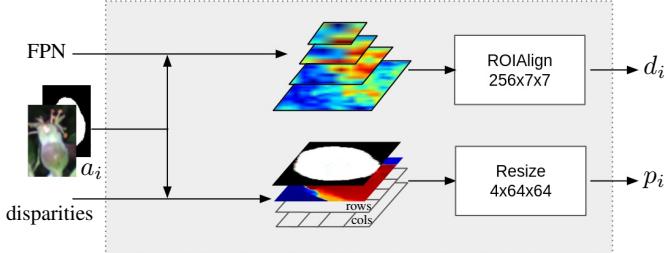


Fig. 8. Local Feature Extractor. The Mask-RCNN feature maps are cropped and passed to ROIAlign to build the visual descriptor. Positional descriptors are built by stacking the bounding box pixel locations with the cropped disparity values and segmentations, and are resized to a fixed shape.

For each  $a_i \in \mathbf{A}^I$ , visual descriptors  $\mathbf{d}_i$  are constructed using the extracted feature maps from Section II-C2 and the ROIAlign [41] operation. ROIAlign is used because it is able to output a feature vector of fixed size regardless of the size of the detected bounding box, while maintaining important feature map information. The feature map at the appropriate pyramid level determined by the bounding box size is cropped and passed to ROIAlign which outputs an  $\mathbb{R}^{256 \times 7 \times 7}$  feature vector. To extract positional descriptors  $\mathbf{p}_i$ , the disparities, segmentations, and x and y pixel locations of the bounding boxes are normalized, stacked, and resized, resulting in an  $\mathbb{R}^{4 \times 64 \times 64}$  positional vector. Disparity values are used to allow the network to reason about 3D information in the image, without requiring the need for camera intrinsics

or to project points onto 3D space. Segmentations provide semantic information regarding which pixels in the bounding box are of greater importance. Our local feature extractor is demonstrated in Fig. 8.

4) *Feature Vector Initialization:* The local features for each detection consist of visual and spatial information across multiple pixels. We use CNNs to both reduce the spatial dimensionality and embed the information into deeper features. This allows the network to simultaneously reason about both appearance and 3D position. The initial node feature vectors are defined as

$${}^{(0)}\mathbf{x}_i = [\text{CNN}_{\text{dec}}(\mathbf{d}_i) + \text{CNN}_{\text{enc}}(\mathbf{p}_i) \parallel s_i \parallel t_i] \quad (2)$$

where  $s_i$  is the Mask-RCNN cluster prediction score from Section II-C2 and allows the network to reason about the importance node  $i$ .  $t_i$  indicates if the node corresponds to a fruitlet  $b_i \in \mathbf{B}^I$  or tag  $\tau$ . Directly injecting this information into the initial feature vector enables the network to better reason about the relationship between fruitlet and tag nodes.

$$t_i = \begin{cases} 1, & \text{if } a_i \equiv \tau \\ 0, & \text{otherwise} \end{cases} \quad (3)$$

5) *Attentional Graph Neural Network:* Drawing inspiration from [50], we build a multiplex graph [58] whose nodes are  $a_i \in \{\mathbf{A}^A \cup \mathbf{A}^B\}$  with initial feature vectors  ${}^{(0)}\mathbf{x}_i \in {}^{(0)}\mathbf{X}$ . The graph has two types of edges: self-edges  $E_{\text{self}}$  that connect all nodes  $i$  to all other nodes from the same image; and cross-edges  $E_{\text{cross}}$  that connect all nodes  $i$  to all nodes from the other image. Message Passing Neural Networks [59] are used to propagate information across edges. For each layer  $\ell \in \{0, 1, \dots, L\}$ , the node feature vectors are updated following

$${}^{(\ell+1)}\mathbf{x}_i = {}^{(\ell)}\mathbf{x}_i + {}^{(\ell)}\text{MLP}([{}^{(\ell)}\mathbf{x}_i \parallel \mathbf{m}_{E \rightarrow i}]) \quad (4)$$

where  $\mathbf{m}_{E \rightarrow i}$  is the aggregation of messages arriving from edges  $\{j : (i, j) \in E\}$  with  $E \in \{E_{\text{self}}, E_{\text{cross}}\}$ . Edges alternate each layer between  $E_{\text{self}}$  and  $E_{\text{cross}}$ , and each layer  $\ell$  has its own learned MLP.

6) *Self-Attentional Aggregation:* Messages are aggregated using the Self-Scaled Dot Product Attention mechanism presented in [60]. Scaled Dot-Product Attention is faster and more space efficient than traditional additive attention, and has demonstrated great success in sequence-based relational tasks. Following the notion of retrieval systems, for each edge  $\{j : (i, j) \in E\}$ , queries  $\mathbf{q}_i$  are mapped against a set of keys  $\mathbf{k}_j$  to assign weight to values  $\mathbf{v}_j$ . The message is then the sum of the weighted values

$$\mathbf{m}_{E \rightarrow i} = \sum_{\{j : (i, j) \in E\}} \alpha_{ij} \mathbf{v}_j \quad (5)$$

where  $\alpha_{ij}$  is the softmax over query-key similarity scores

$$\alpha_{ij} = \text{Softmax}_j(\mathbf{q}_i^T \mathbf{k}_j) \quad (6)$$

The queries, keys, and values are learned linear projections, with each layer having its own projection parameters. Multi-head attention is used as presented in [60]. The final matching descriptors are calculated from learned linear projections

$$\begin{aligned} \mathbf{f}_i^A &= \mathbf{W}^{(L)} \mathbf{x}_i^A + \mathbf{b}, \quad \forall i \in \mathcal{A} \\ \mathbf{f}_j^B &= \mathbf{W}^{(L)} \mathbf{x}_j^B + \mathbf{b}, \quad \forall j \in \mathcal{B} \end{aligned} \quad (7)$$

7) *Optimal Matching Layer*: Once final matching descriptors  $\mathbf{f}_i^A$  and  $\mathbf{f}_j^B$  are calculated, the partial assignment problem must be solved. We use the same optimal transport method as [50] to solve for the partial sum assignment matrix  $\mathbf{P} \in [0, 1]^{M \times N}$ . A score matrix  $\mathbf{S} \in \mathbb{R}^{M \times N}$  is built representing the pairwise score similarity of matching features

$$\mathbf{S}_{ij} = \langle \mathbf{f}_i^A, \mathbf{f}_j^B \rangle, \quad \forall (i, j) \in \mathcal{A} \times \mathcal{B} \quad (8)$$

where  $\langle \cdot, \cdot \rangle$  denotes the inner product. Dustbins are then added to allow for the assignment of unmatched keypoints. The score matrix  $\mathbf{S}$  is augmented to  $\bar{\mathbf{S}} \in \mathbb{R}^{(M+1) \times (N+1)}$  by appending a new row and column filled with a single learnable parameter:

$$\bar{\mathbf{S}}_{i,N+1} = \bar{\mathbf{S}}_{M+1,j} = \bar{\mathbf{S}}_{M+1,N+1} = z \in \mathbb{R} \quad (9)$$

We define the constraints for the augmented partial assignment matrix  $\bar{\mathbf{P}}$  as

$$\begin{aligned} \bar{\mathbf{P}}_{1:N+1} &= [\mathbf{1}_M^T \ N]^T \\ \bar{\mathbf{P}}_{1:M+1} &= [\mathbf{1}_N^T \ M]^T \end{aligned} \quad (10)$$

where  $M$  and  $N$  are appended because each dustbin should have as many matches as there are detections in the other set. We attempt to maximize the total score  $\sum_{(i,j)} \bar{\mathbf{S}}_{i,j} \bar{\mathbf{P}}_{i,j}$  while satisfying the constraints in Equation 10. The above optimization problem is solved using the Sinkhorn algorithm [61], a differentiable version of the Hungarian algorithm [62]. After  $T$  iterations,  $\mathbf{P} = \bar{\mathbf{P}}_{1:M,1:N}$  is recovered and fruitlet nodes  $\mathbf{F}_A$  and  $\mathbf{F}_B$  identified by the Mask-RCNN class outputs are matched.

8) *Loss*: As a result of the Attentional Graph Neural Network and Optimal Matching Layer being fully differentiable, partial assignment predictions can be back propagated all the way to the local features. We use the same negative log-likelihood loss as [50] with ground-truth matched labels  $\mathcal{M} = \{(i, j)\} \subset \mathcal{A} \times \mathcal{B}$  and unmatched labels  $\mathcal{I} \subseteq \mathcal{A}$ ,  $\mathcal{J} \subseteq \mathcal{B}$ , we calculate the partial assignment loss as

$$\begin{aligned} \mathcal{L} = - \sum_{(i,j) \in \mathcal{M}} \log \bar{\mathbf{P}}_{i,j} \\ - \sum_{i \in \mathcal{I}} \log \bar{\mathbf{P}}_{i,N+1} - \sum_{j \in \mathcal{J}} \log \bar{\mathbf{P}}_{M+1,j} \end{aligned} \quad (11)$$

Unlike in [50], where datasets are created by applying homographies and each keypoint is well-defined in either  $M$ ,  $I$ , or  $J$ , our ground-truth labels only consist of fruitlets belonging to the target cluster. Therefore,  $\mathcal{M}$ ,  $\mathcal{I}$ , and  $\mathcal{J}$  only consist of clustered fruitlet correspondences, and as a result not all nodes contribute to the loss.

### III. RESULTS

#### A. Dataset

All of our datasets will be made publicly available at <sup>1</sup>.

<sup>1</sup><https://labs.ri.cmu.edu/aiira/resources/>

1) *Data Collection*: Our dataset consists of stereo-images taken of 252 clusters along with their caliper measurements. The data was collected at the University of Massachusetts Amherst Cold Spring Orchard. The clusters were evenly distributed between three apple varieties of Gala, Honeycrisp, and Fuji. Each cluster was imaged on four different days spread out over an eight day period: 05/18/2021, 05/21/2021, 05/23/2021, and 05/25/2021, which we refer to as Day 1, Day 4, Day 6, and Day 8 respectively. During data collection, the clusters were tagged with an AprilTag and each fruitlet assigned a unique id to track across multiple days. A human manually operated the hand-held stereo camera (Fig. 2), collecting a sequence of images of each cluster (Fig. 9). A single image from each cluster per day was then manually selected and used for sizing evaluation, while the remaining were used to train the detection, segmentation, and association networks. Hand measurements were collected for each fruitlet using a digital caliper (Fig. 10).



Fig. 9. Example subset of an image sequence of a cluster captured in the field. A single image is selected and passed through our computer vision sizing pipeline for evaluation.

Taking measurements with hand calipers naturally results in random errors. This is because the measurement will vary as the caliper is rotated around the fruitlet as a result of its asymmetrical shape. As well, the measurements are dependant on how hard the caliper is closed around the fruitlet, which can easily change on different days. With no exact quantifiable number to express this variation, we asked apple growers how much this measurement can vary. Their response was between 1-1.5mm for two separate measurements of the same fruit on the same day. This is significant as the sizing process specified by the Fruitlet Growth Model begins when fruitlets are as small as 6mm. This further demonstrates the need for alternative solutions.



Fig. 10. Example hand-caliper measurement of a fruitlet.

2) *Annotation Labelling*: To train the Mask-RCNN network, 600 images were labelled with bounding boxes around every fruitlet and polygons around each tag. For the pix2pix network, 300 cropped fruitlets were hand-segmented with a binary mask. Both labelled datasets were divided into training,

TABLE I  
MEAN, MEDIAN, AND STANDARD DEVIATIONS (MM) OF OUR COMPUTER VISION SIZING PIPELINE AND CALIPER MEASUREMENTS

|              | CVSP Mean | CVSP Med | CVSP Std | CM Mean | CM Med | CM Std |
|--------------|-----------|----------|----------|---------|--------|--------|
| <b>Day 1</b> | 6.41      | 6.25     | 1.45     | 5.45    | 5.30   | 1.21   |
| <b>Day 4</b> | 8.18      | 8.07     | 1.63     | 7.58    | 7.40   | 1.68   |
| <b>Day 6</b> | 9.32      | 9.29     | 2.15     | 9.01    | 9.10   | 2.19   |
| <b>Day 8</b> | 9.83      | 9.80     | 2.35     | 9.53    | 9.60   | 2.37   |

validation, and test sets with a 70/15/15 split respectively. When labelling bounding boxes, each fruitlet was classified as either a cluster or non-cluster fruitlet to be used when training Mask-RCNN.

3) *Association Labelling*: To label the data used to train to the fruitlet association network, a custom tool was developed (Fig. 11). Two images of annotated fruitlets and tags on different days are placed side by side, and the user matches the clustered fruitlets by assigning ids. 400 images were labelled to train the network, and datasets were divided into 70/15/15 training, validation, and test splits.



Fig. 11. Custom tool for fruitlet association labelling. (a) Images of the same cluster taken on different days are placed side by side with bounding boxes displayed and clustered fruitlets outlined. (b) The user is able to select and assign matching ids to each fruitlet in the cluster.

Due to the limited size of the dataset, the data was augmented in several ways. Images were randomly flipped horizontally, and labelled bounding boxes were randomly shifted and scaled without requiring the width-height aspect ratio to be maintained. To make the network more robust to various graph structures, in each training batch nodes were randomly dropped from the graph and cluster prediction scores randomly shifted. Augmenting the data had a significant improvement on the performance of our network as demonstrated in Fig. 15.

### B. Fruitlet Sizing

We evaluate our fruitlet sizing approach presented in Section II-B. After all fruitlets are sized, outliers are removed from both the caliper measurements and the outputs from our pipeline using a Z-score threshold of 3. This was necessary to mitigate both substantially incorrect caliper recordings and cases where our trained networks failed to generalize. The number of outliers removed were less than 0.5% of all fruitlets sized. The distribution of measured sizes using our computer vision sizing pipeline (CVSP) and the caliper method (CM) are reported Fig. 12. and Table I. While the computer vision pipeline consistently produces slightly larger results on all days, the growth trends are similar across the 8 day period.

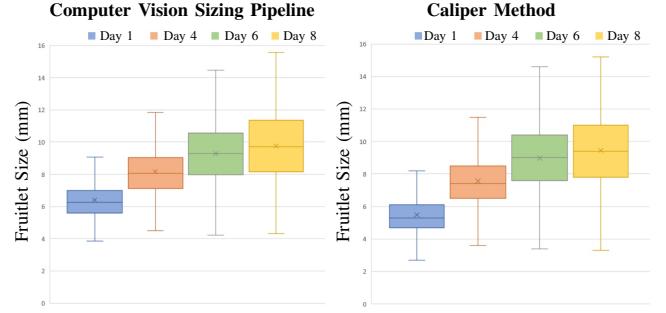


Fig. 12. Distribution of computer vision and caliper method measured fruitlet sizes. The "x" symbol indicates the mean and horizontal line indicates the median.

Table II shows the Mean Absolute Error (MAE) and the Mean Absolute Percentage Error (MAPE) of the computer vision sizes compared against the caliper method for each day. The MAE is just over 1mm on Day 1, and consistently remains under 1mm for the subsequent days. As well, the MAPE is largest on the first day, and reduces as the fruitlets grow, falling under 10% on day 6. One possible reason for this is it is more difficult to accurately measure disparity values and segment smaller fruitlets. On the other hand, the errors may stem from the inconsistencies in measuring ground-truth. The size variations from using calipers would have a more significant effect when the fruitlets are small.

TABLE II  
MAE AND MAPE OF OUR CVSP COMPARED TO CM. MEAN CALIPER  
MEASURED SIZES ARE PROVIDED FOR REFERENCE.

|              | CM Mean (mm) | MAE (mm) | MAPE (%) |
|--------------|--------------|----------|----------|
| <b>Day 1</b> | 5.45         | 1.04     | 21.3     |
| <b>Day 4</b> | 7.58         | 0.749    | 11.1     |
| <b>Day 6</b> | 9.01         | 0.691    | 8.67     |
| <b>Day 8</b> | 9.53         | 0.719    | 8.45     |

The measurements produced by our method have a high correlation with the caliper measurements, with an  $R^2$  score of 0.826 (Fig. 13).

The ultimate goal is to be able to measure fruitlet growth rates to determine when thinning application should be applied. Fruitlets with growth rates less than 50% of the fastest growing fruits are predicted to abscise [16]. Therefore, we evaluate our method's ability to predict the percentage of fruitlets that will abscise and compare the results to the caliper method. According to the Fruitlet Growth Model, fruits should not be measured until they have a diameter of 6mm, and measurements should be spaced three to four days apart. We select the date range of Day 4 to Day 8 because both sizing and timing requirements are satisfied. The median growth

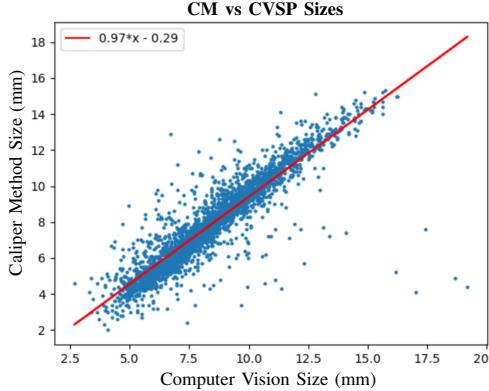


Fig. 13. Linear fit between CM and CVSP measured sizes.

rate of the top 15% fastest growing fruitlets are used to determine the drop percent. The results can be seen in Table III. The computer vision method predicts an almost equivalent percentage of fruitlets will abscise, with less than a 1% difference compared to the caliper method. This is what best demonstrates the effectiveness of our approach. Growers could potentially draw the same conclusions about when to spray using our system as they would using the current method, and without the need to manually size each fruitlet. Fig. 14 shows the distributions of growth rates over the date range used. The distributions follow similar trends.

TABLE III

EVALUATION OF GROWTH RATES MEASURED USING CVSP AND CM.  
ABSCISE PERCENT (AP) IS CALCULATED USING THE MEDIAN OF THE  
GROWTH RATES OF THE TOP 15% FASTEST GROWING FRUITLETS (MFG).

|         | CVSP MFG | CM MFG | CVSP AP | CM AP |
|---------|----------|--------|---------|-------|
| Day 4-8 | 3.76mm   | 4.00mm | 55.6%   | 54.8% |

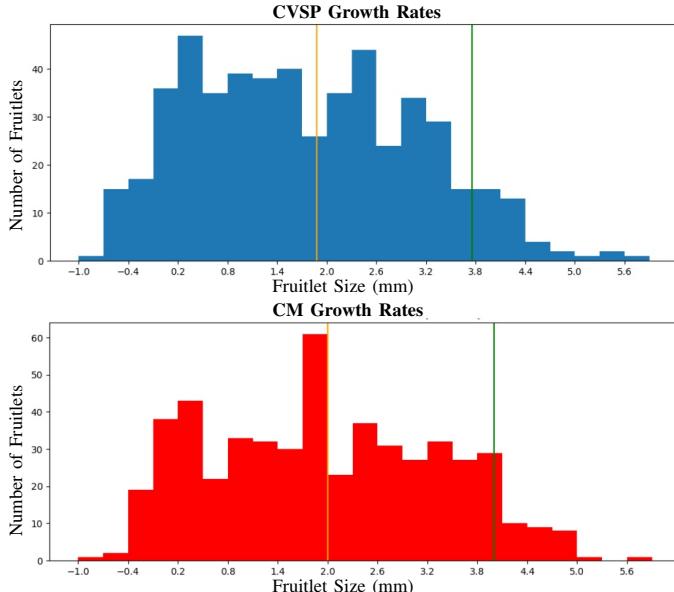


Fig. 14. Distribution of growth rates over Day 4-8. The green bar represents the median growth of the top 15% fastest growing fruits. The orange bar indicates 50% of this value which is used to calculate the abscise percentage.

We assess the speed of our pipeline. Table IV shows the distribution of sizing times from image input to sizing output.

We ran our evaluation on an NVIDIA GeForce RTX 3070 GPU. The average processing time is approximately 4.34s, with RAFT-Stereo consuming a majority of the time with an average of 4.05s. The speed of RAFT-Stereo and hence our pipeline will be affected by the GPU used.

TABLE IV  
RUNTIMES OF DIFFERENT CVSP MODULES.

|                    | Mean (s) | Med (s) | Std (s) |
|--------------------|----------|---------|---------|
| <b>RAFT-Stereo</b> | 4.05     | 4.03    | 0.144   |
| <b>Mask-RCNN</b>   | 0.205    | 0.203   | 0.0198  |
| <b>pix2pix</b>     | 0.0683   | 0.0650  | 0.0400  |
| <b>Sizing</b>      | 0.0189   | 0.0184  | 0.00686 |
| <b>Total</b>       | 4.34     | 4.32    | 0.152   |

We asked apple growers how long it usually takes to size fruitlets with calipers. The response we received is that it requires a minimum of 30s per cluster. Based on this preliminary study, our computer vision pipeline was able to size fruitlets 7 times faster compared to hand-caliper measurements. We plan to evaluate this performance improvement at a larger scale in the future.

### C. Cross-Day Fruitlet Association

We evaluate the performance of our cross-day fruitlet association network. The precision, recall, and matching scores are computed against the partial assignment matching threshold and reported in Fig. 15. Matching score is the average ratio of correct associations over the total number of fruitlets belonging to the target cluster. Our network achieves strong performance, with a matching score of 95.1% at a matching threshold up to 0.8. Qualitative examples can be seen in Fig. 19.

We also run ablation tests to prove the validity of our design choices. The matching scores with varying match thresholds are reported for the following experiments:

- i *Cross-Day Fruitlet Association*: Our primary GNN network presented in Section II-C5.
- ii *Simple Positional Descriptor*: Segmentation and disparity information is removed from the positional descriptor  $p_i$  leaving only pixel row and column information.
- iii *No Tag Feature Initialization*: The tag information  $t_i$  is not concatenated onto the initial feature vector (Equation 2).
- iv *No Score Feature Initialization*: The score information  $s_i$  is not concatenated onto the initial feature vector (Equation 2).
- v *No Data Augmentation*: The data augmentation methods described in Section III-A3 are removed from training.

As seen in Fig. 15, each of our design choices has a positive effect on matching score. While embedding disparity and segmentation into the positional descriptor leads to the greatest improvement, data augmentation and directly injecting classification score and tag information into the node feature vector all provide additional value, ultimately resulting in a matching score above 95%.

### D. Automated Growth Tracking

We evaluate growth rates measured using a full end-to-end fruitlet growth measurement pipeline (FGMP) and the

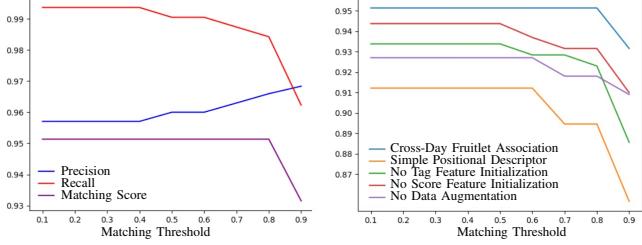


Fig. 15. Left: precision, recall, and matching score for the cross-day fruitlet association network. Right: ablation study of our cross-day fruitlet association network. Our presented network achieves the highest matching score.

caliper method across the Day 4 to Day 8 date range. The growth measurement pipeline consists of both fruitlet sizing and association, for a fully automated growth tracking system with no manual fruitlet identification. Outliers are removed from both measurements using a Z-score threshold of 3, with the number of outliers being less than 2.5% of the total measured growth rates. The results are shown in Table V and Fig. 16. Our end-to-end pipeline predicts approximately the same number of fruitlets will abscise compared to the caliper method, with less than a 3% difference. Apple growers would be able to draw similar conclusions about when to spray using our automated approach that does not require any manual sizing or fruitlet identification.

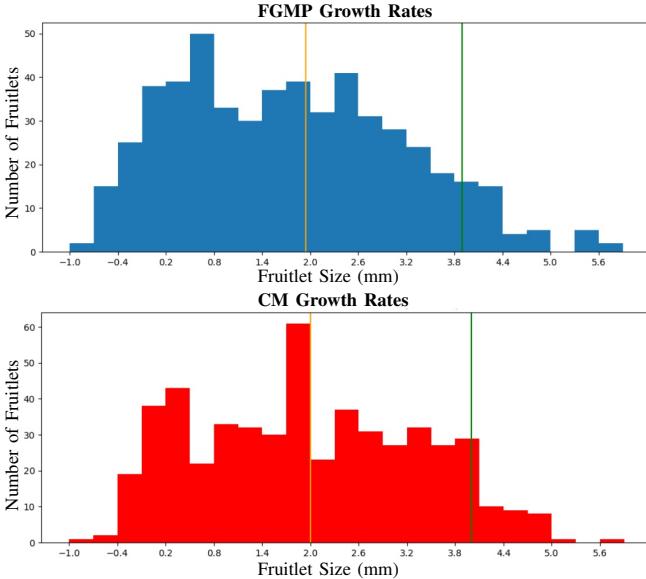


Fig. 16. Distribution of growth rates over Day 4-8. The green bar represents the median growth of the top 15% fastest growing fruits. The orange bar indicates 50% of this value which is used to calculate the abscise percentage.

TABLE V  
EVALUATION OF GROWTH RATES MEASURED USING FGMP AND CM.  
ABSCISE PERCENT (AP) IS CALCULATED USING THE MEDIAN OF THE  
GROWTH RATES OF THE TOP 15% FASTEST GROWING FRUITLETS (MFG).

|                | <b>FGMP MFG</b> | <b>CM MFG</b> | <b>FGMP AP</b> | <b>CM AP</b> |
|----------------|-----------------|---------------|----------------|--------------|
| <b>Day 4-8</b> | 3.89mm          | 4.00mm        | 57.6%          | 54.8%        |

The association network took an average of 0.570s to process a pair of images with a standard deviation of 0.0224s.

### E. Robot Experiment

We test our sizing pipeline on images collected by a robot in the field. The flash stereo camera is attached to a 7 DoF robotic arm consisting of a UR5 and linear slider [63]. Images of 20 clusters were taken at the University of Massachusetts Amherst Cold Spring Orchard on four different days spanning a five day window: 05/22/2022, 05/24/2022, 05/25/2022, and 05/26/2022, which we denote as Day 1, Day 3, Day 4, and Day 5 respectively. The robot was controlled to follow a predefined semi-spherical motion path around the detected AprilTag, collecting images at viewpoints equally distributed throughout the motion path (Fig. 17). A single image of each cluster for each day was selected and used for evaluation.

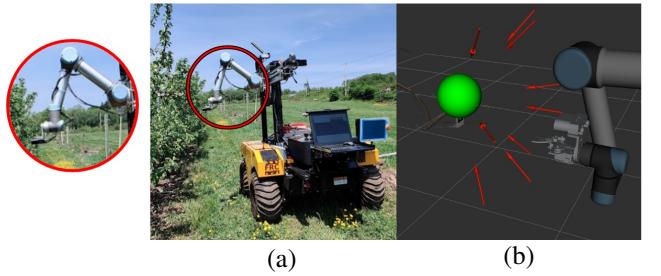


Fig. 17. (a) Flash stereo camera attached to 7 DoF robotic arm. (b) Robot motion path where 15 images are collected at evenly distributed viewpoints on a semi-sphere around the detected cluster.

We record the size distributions (Fig. 18) for measurements taken by both our computer vision sizing pipeline with robot captured images (RCVSP) and the caliper method (CM). On Day 1, the sizes produced by the computer vision pipeline are larger than those produced by the caliper method; however, as the days progress the caliper measurements are larger and the difference increases. This is because the robot is naively following a pre-set path and not reasoning about where to capture an image. As a result, smaller fruits are more likely to be completely occluded, and larger fruits partially occluded. This biases the measurements produced by our system to more accurately size medium-sized fruitlets, as smaller fruits are more likely to be missed and larger fruits undersized.

TABLE VI  
MEAN AND MEDIAN RCVSP AND CM MEASUREMENTS (MM).

|              | <b>RCVSP Mean</b> | <b>RCVSP Med</b> | <b>CM Mean</b> | <b>CM Med</b> |
|--------------|-------------------|------------------|----------------|---------------|
| <b>Day 1</b> | 6.68              | 6.47             | 6.58           | 6.20          |
| <b>Day 3</b> | 6.96              | 6.82             | 7.03           | 6.90          |
| <b>Day 4</b> | 7.17              | 6.97             | 7.31           | 7.00          |
| <b>Day 5</b> | 7.35              | 7.19             | 7.49           | 7.20          |

The MAE, MAPE, and Mean Difference between the RCVSP and CM sizes are presented in Table VII. We also report the percentage of fruitlets that could not be sized by RCVSP as a result of occlusions or missed detections. The most notable difference between these results and the ones demonstrated by the hand-held camera (Section III-B) is that the MAPE scores remain relatively constant instead of decreasing as the fruitlets grow. This is once again a result of biased results towards medium sized fruits. Smaller fruitlets are more likely to be fully occluded or undetected, resulting

TABLE VII

MAE, MAPE, AND MEAN DIFFERENCE OF RCVSP COMPARED TO CM. MEAN CM SIZES ARE PROVIDED FOR REFERENCE. THE PERCENTAGE OF FRUITLETS UNABLE TO BE SIZED BY THE ROBOT, EITHER DUE TO FAILED DETECTIONS OR OCCLUSIONS, IS REPORTED.

|              | <b>CM Mean (mm)</b> | <b>MAE (mm)</b> | <b>MAPE (%)</b> | <b>Mean Difference (mm)</b> | <b>Not Sized %</b> |
|--------------|---------------------|-----------------|-----------------|-----------------------------|--------------------|
| <b>Day 1</b> | 6.58                | 0.811           | 12.4            | 0.101                       | 14.7               |
| <b>Day 3</b> | 7.03                | 0.891           | 12.7            | -0.0762                     | 11.0               |
| <b>Day 4</b> | 7.31                | 0.858           | 12.0            | -0.144                      | 7.33               |
| <b>Day 5</b> | 7.49                | 0.925           | 13.1            | -0.145                      | 11.0               |

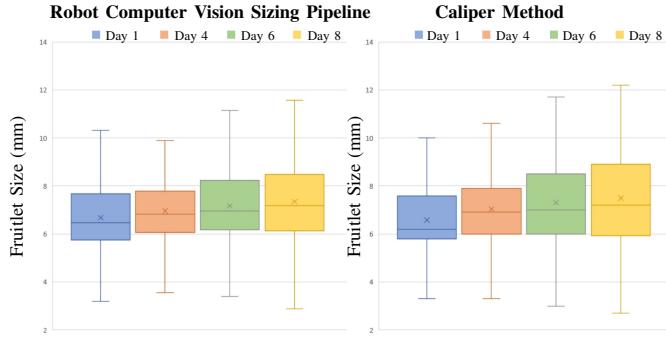


Fig. 18. Distribution of RCVSP and CM measured fruitlet sizes. The "x" symbol indicates the mean and the horizontal line the median.

in a greater not sized percentage and Mean Difference on Day 1. Larger and partially occluded fruits are more likely to be undersized, resulting in decreasing Mean Differences as the days progress. The smallest Mean Difference occurs when the fruitlet is around 7mm.

#### IV. CONCLUSION

This paper presents an alternative approach to sizing and measuring the growth rates of apple fruitlets. We have demonstrated that our computer vision-based method is able to produce similar results as the current caliper method used in practice. Most notably, we are able to predict similar abscise rates with only a single stereo image pair per cluster, without any human effort required to label fruitlets or take caliper measurements. The advantage our system brings is a faster and less labour intensive approach that produces comparable results.

While our approach produces promising results, there is still work needed in order to make it adoptable by growers. For one, it requires the use of a stereo camera that is able to take quality images in light varying environments. We used a custom-made illumination invariant camera system [32] that is not freely available. As well, to process results in real-time, a computationally sufficient device has to be carried out in the field and connected to the camera. This brings challenges as the wiring makes it difficult to maneuver around the cluster. In an ideal scenario, lightweight models small enough to run mobile devices would be used. While it is possible to replace the backbone of Mask-RCNN with lightweight networks [64], [65], RAFT-Stereo still requires sufficient computational resources. Further investigation is needed to determine if faster learning-based stereo matching algorithms [66], [67] provide sufficient performance to be used in agriculture. Alternatively, one advantage to our fully automated approach is images

can be collected and processed offline. This would require connecting the camera to a device with sufficient memory capacity to save the captured images to be processed at a future point in time.

The next step to make the process fully autonomous is to replace the need for humans to collect images. This paper evaluates the performance of images collected by a robot following a set motion path. This process itself is not fully autonomous, as a human is needed to select the best quality images to run through the sizing pipeline. The results are slightly biased towards accurate sizing of medium-sized fruit. Smaller fruits are more fully occluded and cannot be detected, and larger fruits are more partially occluded and undersized. This is because the robot is not reasoning about the environment as it captures images, unlike a human is with the hand-held camera. While one possible solution to handle partial occlusions is to incorporate the work of Dong *et al.* [68] to directly infer ellipses, this would require labelling copious amounts of training data that would likely have to be repeated across datasets. As well, this would not handle the case of fully occluded fruits.

In our future work, we focus on integrating next best view optimization for the task of fully automated fruitlet sizing. This will allow the robot to be able to take high quality images of all fruitlets in the cluster. Recently, there has been work dedicated towards next best view optimization for fruit sizing using region of interest exploration [21], [69], [70], but these approaches rely on low resolution 3D maps that are insufficient to size fruitlets. As well, sizing is performed by fitting 3D shapes, which is challenging for apple fruitlets because of their small size and often only having one side of their surfaces visible as a result of the occluded environment. We will build upon these works by adapting next best view planning techniques to the fruitlet domain and by integrating both 3D and 2D sizing techniques.

#### ACKNOWLEDGMENTS

We would like to thank the University of Massachusetts Amherst Cold Spring Orchard for allowing us to collect data. This research was funded by NSF / USDA NIFA 2020-01469-1022394.

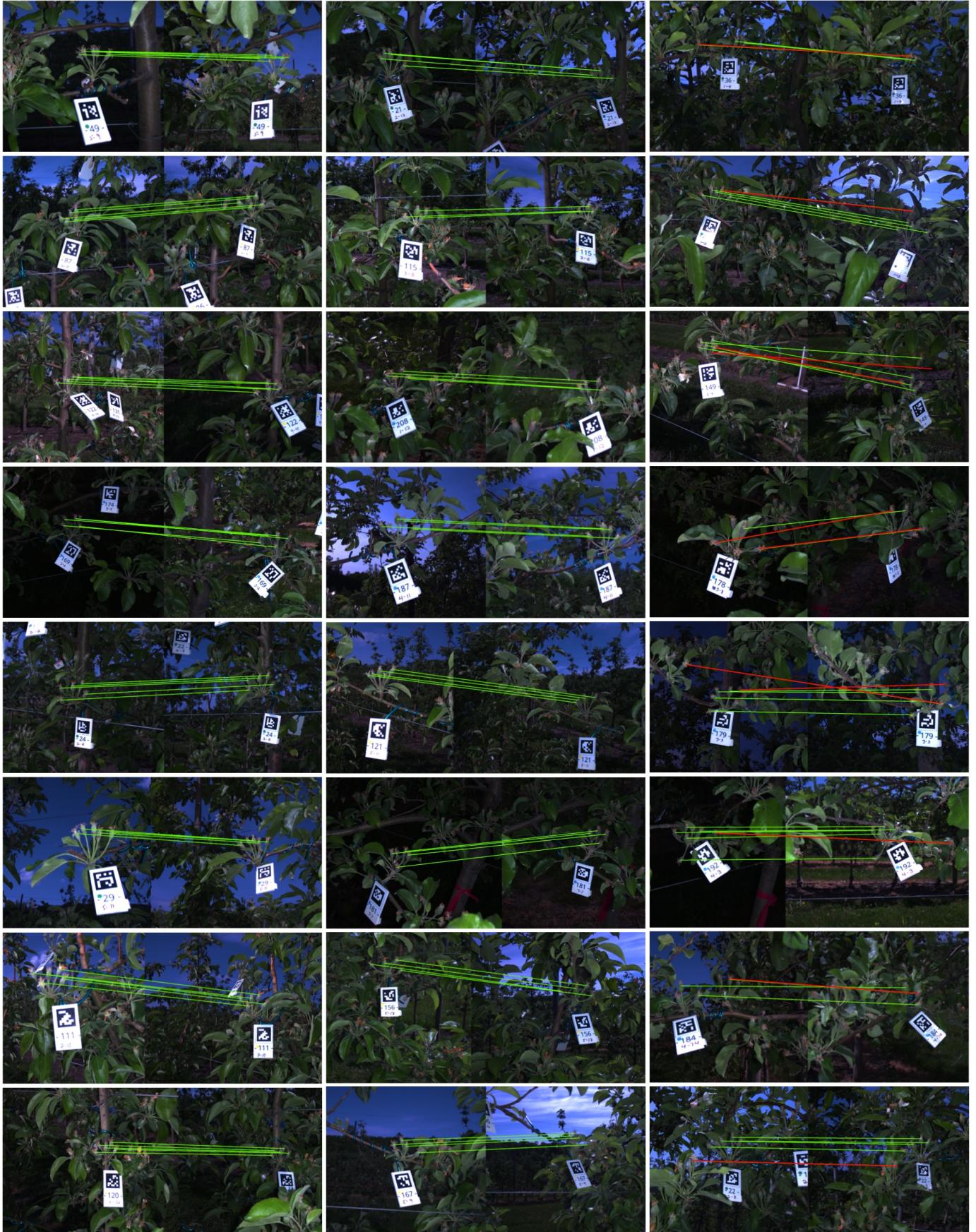


Fig. 19. Examples of cross-day fruitlet association results. Left column: correctly associated fruitlets. Middle column: correctly associated fruitlets when a fruitlet is either occluded or fallen off. Right: incorrect association examples.

## REFERENCES

- [1] D. Font, T. Pallejà, M. Tresanchez, D. Runcan, J. Moreno, D. Martínez, M. Teixidó, and J. Palacín, "A proposal for automatic fruit harvesting by combining a low cost stereovision camera and a robotic arm," *Sensors*, vol. 14, no. 7, pp. 11 557–11 579, 2014. [Online]. Available: <https://www.mdpi.com/1424-8220/14/7/11557>
- [2] C. McCool, I. Sa, F. Dayoub, C. Lehnert, T. Perez, and B. Upcroft, "Visual detection of occluded crop: For automated harvesting," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 2506–2512.
- [3] S. Paulus, J. Behmann, A.-K. Mahlein, L. Plümer, and H. Kuhlmann, "Low-cost 3d systems: Suitable tools for plant phenotyping," *Sensors*, vol. 14, no. 2, pp. 3001–3018, 2014. [Online]. Available: <https://www.mdpi.com/1424-8220/14/2/3001>
- [4] A. L. Chandra, S. V. Desai, W. Guo, and V. N. Balasubramanian, "Computer vision with deep learning for plant phenotyping in agriculture: A survey," *CoRR*, vol. abs/2006.11391, 2020. [Online]. Available: <https://arxiv.org/abs/2006.11391>
- [5] S. T. Nuske, K. Wilshusen, S. Achard, L. Yoder, S. G. Narasimhan, and S. Singh, "Automated visual yield estimation in vineyards," *J. Field Robotics*, vol. 31, pp. 837–860, 2014.
- [6] H. Cheng, L. Damerow, Y. Sun, and M. Blanke, "Early yield prediction using image analysis of apple fruit and tree canopy features with neural networks," *Journal of Imaging*, vol. 3, no. 1, 2017. [Online]. Available: <https://www.mdpi.com/2313-433X/3/1/6>
- [7] B. Darwin, P. Dharmaraj, S. Prince, D. E. Popescu, and D. J. Hemanth, "Recognition of bloom/yield in crop images using deep learning models for smart agriculture: A review," *Agronomy*, vol. 11, no. 4, 2021. [Online]. Available: <https://www.mdpi.com/2073-4395/11/4/646>
- [8] M. Stein, S. Bargoti, and J. Underwood, "Image based mango fruit detection, localisation and yield estimation using multiple view geometry," *Sensors*, vol. 16, no. 11, p. 1915, 2016.
- [9] X. Liu, S. W. Chen, C. Liu, S. S. Shivakumar, J. Das, C. J. Taylor, J. Underwood, and V. Kumar, "Monocular camera based fruit counting and mapping with semantic data association," *IEEE Robotics and Automation Letters*, vol. 4, no. 3, pp. 2296–2303, 2019.
- [10] W. Qureshi, A. Payne, K. Walsh, R. Linker, O. Cohen, and M. Dailey, "Machine vision for counting fruit on mango tree canopies," *Precision Agriculture*, vol. 17, 04 2017.
- [11] A. K. Nellithimaru and G. A. Kantor, "Rols: Robust object-level slam for grape counting," in *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition Workshops*, 2019, pp. 0–0.
- [12] A. Janowski, R. Kaźmierczak, C. Kowalczyk, and J. Szulwic, "Detecting apples in the wild: Potential for harvest quantity estimation," *Sustainability*, vol. 13, no. 14, 2021. [Online]. Available: <https://www.mdpi.com/2071-1050/13/14/8054>
- [13] H. Kang, H. Zhou, and C. Chen, "Visual perception and modeling for autonomous apple harvesting," *IEEE Access*, vol. 8, pp. 62 151–62 163, 2020.
- [14] Y. Ge, Y. Xiong, and P. J. From, "Symmetry-based 3d shape completion for fruit localisation for harvesting robots," *Biosystems Engineering*, vol. 197, pp. 188–202, 2020. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S1537511020301963>
- [15] V. R. Vijaykumar and S. Vanik, "Agricultural robot: Leaf disease detection and monitoring the field condition using machine learning and image processing," 2018.
- [16] D. W. Greene, A. N. Lakso, T. L. Robinson, and P. Schwallier, "Development of a fruitlet growth model to predict thinner response on apples," *HortScience*, vol. 48, no. 5, pp. 584–587, 2013.
- [17] Z. Wang, A. Koirlala, K. Walsh, N. Anderson, and B. Verma, "In field fruit sizing using a smart phone application," *Sensors*, vol. 18, no. 10, 2018. [Online]. Available: <https://www.mdpi.com/1424-8220/18/10/3331>
- [18] Y. Wang and Y. Chen, "Fruit morphological measurement based on three-dimensional reconstruction," *Agronomy*, vol. 10, no. 4, 2020. [Online]. Available: <https://www.mdpi.com/2073-4395/10/4/455>
- [19] T. Jadhav, K. Singh, and A. Abhyankar, "Volumetric estimation using 3d reconstruction method for grading of fruits," *Multimedia Tools and Applications*, vol. 78, pp. 1613–1634, 2018.
- [20] C. Lehnert, I. Sa, C. McCool, B. Upcroft, and T. Perez, "Sweet pepper pose detection and grasping for automated crop harvesting," in *2016 IEEE International Conference on Robotics and Automation (ICRA)*, 2016, pp. 2428–2434.
- [21] S. Marangoz, T. Zaenker, R. Menon, and M. Bennewitz, "Fruit mapping with shape completion for autonomous crop monitoring," 2022. [Online]. Available: <https://arxiv.org/abs/2203.15489>
- [22] A. Gongal, M. Karkee, and S. Amatya, "Apple fruit size estimation using a 3d machine vision system," *Information Processing in Agriculture*, vol. 5, no. 4, pp. 498–503, 2018. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S2214317317302408>
- [23] E. Grilli, R. Battisti, and F. Remondino, "An advanced photogrammetric solution to measure apples," *Remote Sensing*, vol. 13, no. 19, 2021. [Online]. Available: <https://www.mdpi.com/2072-4292/13/19/3960>
- [24] Z. Wang, K. B. Walsh, and B. Verma, "On-tree mango fruit size estimation using rgb-d images," *Sensors*, vol. 17, no. 12, 2017. [Online]. Available: <https://www.mdpi.com/1424-8220/17/12/2738>
- [25] M. Stein, S. Bargoti, and J. Underwood, "Image based mango fruit detection, localisation and yield estimation using multiple view geometry," *Sensors*, vol. 16, no. 11, 2016. [Online]. Available: <https://www.mdpi.com/1424-8220/16/11/1915>
- [26] J. M. Ponce, A. Aquino, B. Millan, and J. M. Andújar, "Automatic counting and individual size and mass estimation of olive-fruits through computer vision techniques," *IEEE Access*, vol. 7, pp. 59 451–59 465, 2019.
- [27] W. Zhang, J. Wang, Y. Liu, K. Chen, H. Li, Y. Duan, W. Wu, Y. Shi, and W. Guo, "Deep-learning-based in-field citrus fruit detection and tracking," *Horticulture Research*, vol. 9, 02 2022, uhac003. [Online]. Available: <https://doi.org/10.1093/hr/uhac003>
- [28] A. Gongal, A. Silwal, S. Amatya, M. Karkee, Q. Zhang, and K. Lewis, "Apple crop-load estimation with over-the-row machine vision system," *Computers and Electronics in Agriculture*, vol. 120, pp. 26–35, 2016. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S016816991500335X>
- [29] T. Hondo, K. Kobayashi, and Y. Aoyagi, "Real-time prediction of growth characteristics for individual fruits using deep learning," *Sensors*, vol. 22, no. 17, 2022. [Online]. Available: <https://www.mdpi.com/1424-8220/22/17/6473>
- [30] M. Gori, G. Monfardini, and F. Scarselli, "A new model for learning in graph domains," in *Proceedings. 2005 IEEE International Joint Conference on Neural Networks, 2005.*, vol. 2, 2005, pp. 729–734 vol. 2.
- [31] E. Olson, "Apriltag: A robust and flexible visual fiducial system," in *2011 IEEE International Conference on Robotics and Automation*, 2011, pp. 3400–3407.
- [32] A. Silwal, T. Parhar, F. Yandun, H. Bawea, and G. Kantor, "A robust illumination-invariant camera system for agricultural applications," in *2021 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, 2021, pp. 3292–3298.
- [33] N. Tsoulias, D. S. Paraforos, G. Xanthopoulos, and M. Zude-Sasse, "Apple shape detection based on geometric and radiometric features using a lidar laser scanner," *Remote Sensing*, vol. 12, no. 15, 2020. [Online]. Available: <https://www.mdpi.com/2072-4292/12/15/2481>
- [34] J. P. Underwood, C. Hung, B. Whelan, and S. Sukkarieh, "Mapping almond orchard canopy volume, flowers, fruit and yield using lidar and vision sensors," *Computers and Electronics in Agriculture*, vol. 130, pp. 83–96, 2016. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0168169916308249>
- [35] U. Weiss and P. Biber, "Plant detection and mapping for agricultural robots using a 3d lidar sensor," *Robotics and Autonomous Systems*, vol. 59, no. 5, pp. 265–273, 2011, special Issue ECML 2009. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0921889011000315>
- [36] M. Qadri, "Robotic vision for 3d modeling and sizing in agriculture," Master's thesis, Carnegie Mellon University, Pittsburgh, PA, August 2021.
- [37] A. Tonioni, F. Tosi, M. Poggi, S. Mattoccia, and L. Di Stefano, "Real-time self-adaptive deep stereo," 2018. [Online]. Available: <https://arxiv.org/abs/1810.05424>
- [38] L. Lipson, Z. Teed, and J. Deng, "RAFT-Stereo: Multilevel Recurrent Field Transforms for Stereo Matching," *arXiv preprint arXiv:2109.07547*, 2021.
- [39] H. Hirschmüller, "Stereo processing by semiglobal matching and mutual information," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 2, pp. 328–341, 2008.
- [40] S. Ren, K. He, R. Girshick, and J. Sun, "Faster r-cnn: Towards real-time object detection with region proposal networks," 2016.
- [41] K. He, G. Gkioxari, P. Dollár, and R. B. Girshick, "Mask R-CNN," *CoRR*, vol. abs/1703.06870, 2017. [Online]. Available: <https://arxiv.org/abs/1703.06870>
- [42] Y. Wu, A. Kirillov, F. Massa, W.-Y. Lo, and R. Girshick, "Detectron2," <https://github.com/facebookresearch/detectron2>, 2019.
- [43] P. Isola, J.-Y. Zhu, T. Zhou, and A. A. Efros, "Image-to-image translation with conditional adversarial networks," 2018.

- [44] R. Barth, J. IJsselmuider, J. Hemming, and E. V. Henten, “Data synthesis methods for semantic segmentation in agriculture: A capsicum annum dataset,” *Computers and Electronics in Agriculture*, vol. 144, pp. 284–296, 2018. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0168169917305689>
- [45] T. Parhar, H. Baweja, M. Jenkins, and G. Kantor, “A deep learning-based stalk grasping pipeline,” in *2018 IEEE International Conference on Robotics and Automation (ICRA)*, 2018, pp. 6161–6167.
- [46] J. Olatunji, G. Redding, C. Rowe, and A. East, “Reconstruction of kiwifruit fruit geometry using a cgan trained on a synthetic dataset,” *Computers and Electronics in Agriculture*, vol. 177, p. 105699, 2020. [Online]. Available: <https://www.sciencedirect.com/science/article/pii/S0168169920310206>
- [47] G. Bradski, “The OpenCV Library,” *Dr. Dobb's Journal of Software Tools*, 2000.
- [48] Y. Wang, X. Weng, and K. Kitani, “Joint detection and multi-object tracking with graph neural networks,” *CoRR*, vol. abs/2006.13164, 2020. [Online]. Available: <https://arxiv.org/abs/2006.13164>
- [49] M. Buchner and A. Valada, “3d multi-object tracking using graph neural networks with cross-edge modality attention,” 2022. [Online]. Available: <https://arxiv.org/abs/2203.10926>
- [50] P. Sarlin, D. DeTone, T. Malisiewicz, and A. Rabinovich, “Superglue: Learning feature matching with graph neural networks,” *CoRR*, vol. abs/1911.11763, 2019. [Online]. Available: <http://arxiv.org/abs/1911.11763>
- [51] Y. Shi, J.-X. Cai, Y. Shavit, T.-J. Mu, W. Feng, and K. Zhang, “Clustergnn: Cluster-based coarse-to-fine graph neural network for efficient feature matching,” 2022. [Online]. Available: <https://arxiv.org/abs/2204.11700>
- [52] T. N. Kipf and M. Welling, “Semi-supervised classification with graph convolutional networks,” *CoRR*, vol. abs/1609.02907, 2016. [Online]. Available: <http://arxiv.org/abs/1609.02907>
- [53] W. L. Hamilton, R. Ying, and J. Leskovec, “Inductive representation learning on large graphs,” *CoRR*, vol. abs/1706.02216, 2017. [Online]. Available: <http://arxiv.org/abs/1706.02216>
- [54] C. Morris, M. Ritze, M. Fey, W. L. Hamilton, J. E. Lenssen, G. Rattan, and M. Grohe, “Weisfeiler and leman go neural: Higher-order graph neural networks,” *CoRR*, vol. abs/1810.02244, 2018. [Online]. Available: <http://arxiv.org/abs/1810.02244>
- [55] P. Veličković, G. Cucurull, A. Casanova, A. Romero, P. Liò, and Y. Bengio, “Graph attention networks,” 2017. [Online]. Available: <https://arxiv.org/abs/1710.10903>
- [56] K. He, X. Zhang, S. Ren, and J. Sun, “Deep residual learning for image recognition,” 2015. [Online]. Available: <https://arxiv.org/abs/1512.03385>
- [57] T. Lin, P. Dollár, R. B. Girshick, K. He, B. Hariharan, and S. J. Belongie, “Feature pyramid networks for object detection,” *CoRR*, vol. abs/1612.03144, 2016. [Online]. Available: <http://arxiv.org/abs/1612.03144>
- [58] P. J. Mucha, T. Richardson, K. Macon, M. A. Porter, and J.-P. Onnela, “Community structure in time-dependent, multiscale, and multiplex networks,” *Science*, vol. 328, no. 5980, pp. 876–878, may 2010. [Online]. Available: <https://doi.org/10.1126%2Fscience.1184819>
- [59] J. Gilmer, S. S. Schoenholz, P. F. Riley, O. Vinyals, and G. E. Dahl, “Neural message passing for quantum chemistry,” *CoRR*, vol. abs/1704.01212, 2017. [Online]. Available: <http://arxiv.org/abs/1704.01212>
- [60] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. Kaiser, and I. Polosukhin, “Attention is all you need,” *CoRR*, vol. abs/1706.03762, 2017. [Online]. Available: <http://arxiv.org/abs/1706.03762>
- [61] M. Cuturi, “Sinkhorn distances: Lightspeed computation of optimal transportation distances,” 2013. [Online]. Available: <https://arxiv.org/abs/1306.0895>
- [62] J. R. Munkres, “Algorithms for the Assignment and Transportation Problems,” *Journal of the Society for Industrial and Applied Mathematics*, vol. 5, no. 1, pp. 32–38, March 1957.
- [63] A. Silwal, F. Yandún, A. K. Nellithimaru, T. Bates, and G. Kantor, “Bumblebee: A path towards fully autonomous robotic vine pruning,” *CoRR*, vol. abs/2112.00291, 2021. [Online]. Available: <https://arxiv.org/abs/2112.00291>
- [64] A. G. Howard, M. Zhu, B. Chen, D. Kalenichenko, W. Wang, T. Weyand, M. Andreetto, and H. Adam, “Mobilennets: Efficient convolutional neural networks for mobile vision applications,” 2017. [Online]. Available: <https://arxiv.org/abs/1704.04861>
- [65] H. Li, A. Wu, W. Fang, Q. Zhang, M. Liu, Q. Liu, and W. Chen, “Lightweight mask r-cnn for long-range wireless power transfer sys-
- tems,” in *2019 11th International Conference on Wireless Communications and Signal Processing (WCSP)*, 2019, pp. 1–6.
- [66] J.-R. Chang, P.-C. Chang, and Y.-S. Chen, “Attention-aware feature aggregation for real-time stereo matching on edge devices,” in *Proceedings of the Asian Conference on Computer Vision (ACCV)*, November 2020.
- [67] V. Tankovich, C. Häne, Y. Zhang, A. Kowdle, S. Fanello, and S. Bouaziz, “Hitnet: Hierarchical iterative tile refinement network for real-time stereo matching,” 2020. [Online]. Available: <https://arxiv.org/abs/2007.12140>
- [68] W. Dong, P. Roy, C. Peng, and V. Isler, “Ellipse r-CNN: Learning to infer elliptical object from clustering and occlusion,” *IEEE Transactions on Image Processing*, vol. 30, pp. 2193–2206, 2021. [Online]. Available: <https://doi.org/10.1109%2Fitp.2021.3050673>
- [69] T. Zaenker, C. Smitt, C. McCool, and M. Bennewitz, “Viewpoint planning for fruit size and position estimation,” *CoRR*, vol. abs/2011.00275, 2020. [Online]. Available: <https://arxiv.org/abs/2011.00275>
- [70] X. Zeng, T. Zaenker, and M. Bennewitz, “Deep reinforcement learning for next-best-view planning in agricultural applications,” in *2022 International Conference on Robotics and Automation (ICRA)*, 2022, pp. 2323–2329.